Computer Architecture: Science or Religion

David B. Nelson, Ph.D.
Director
National Coordination Office for
Information Technology Research and Development

Commodity Cluster Symposium

July 23, 2003

National Coordination Office (NCO) for Information Technology Research and Development (IT R&D)

Mission: *To formulate and promote Federal information technology research and development to meet national goals.*

- NCO Director reports to the Director of the White House Office of Science Technology Policy (OSTP) and co-chairs the Interagency Working Group for IT R&D
- Coordinates planning, budget, and assessment activities for the Federal multi-agency Networking and Information Technology R&D (NITRD) Program
- Supports six technical Coordinating Groups (CGs) that report to the Interagency Working Group

NITRD Program Coordination

WHITE HOUSE

Executive Office of the President
Office of Science and Technology Policy

National Science and Technology Council

President's Information Technology Advisory Committee (PITAC)

U.S. Congress

NITRD Authorization and Appropriations Legislation

National Coordination Office (NCO) for Information Technology Research and Development

Interagency Working Group (IWG) on Information Technology R&D

Participating Agencies: AHRQ, DARPA, DOE/NNSA, DOE/SC, EPA, NASA, NIH, NIST, NOAA, NSA, NSF, ODDR&E

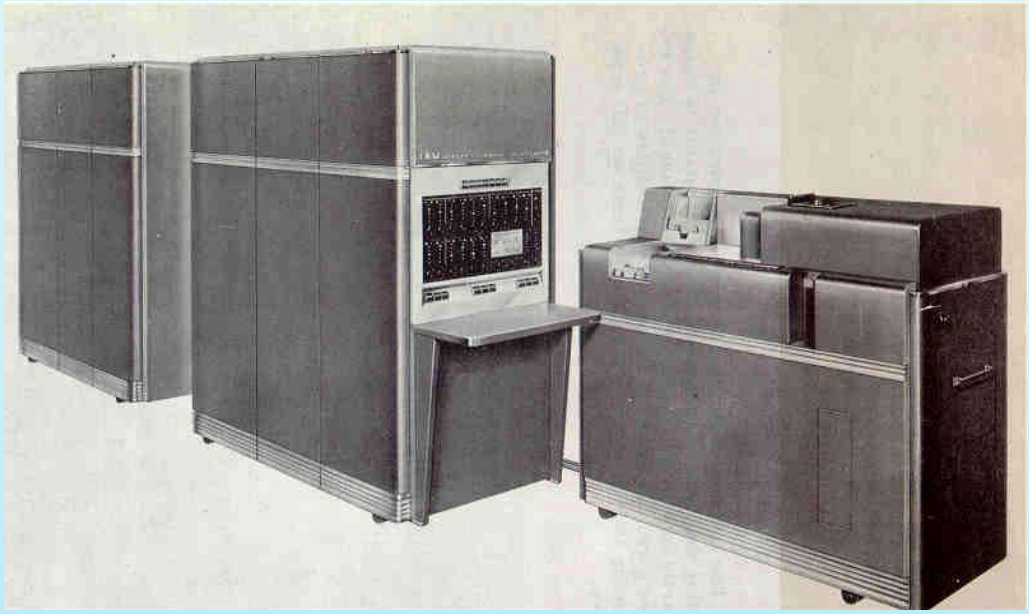
- **High End Computing Coordinating Group (HEC)**
- **Large Scale Networking Coordinating Group (LSN)**
- **High Confidence Software and Systems Coordinating Group (HCSS)**
- **Human Computer Interaction & Information Management Coordinating Group (HCI & IM)**
- **Software Design and Productivity Coordinating Group (SDP)**
- **Social, Economic and Workforce Implications of IT and IT Workforce Development Coordinating Group (SEW)**

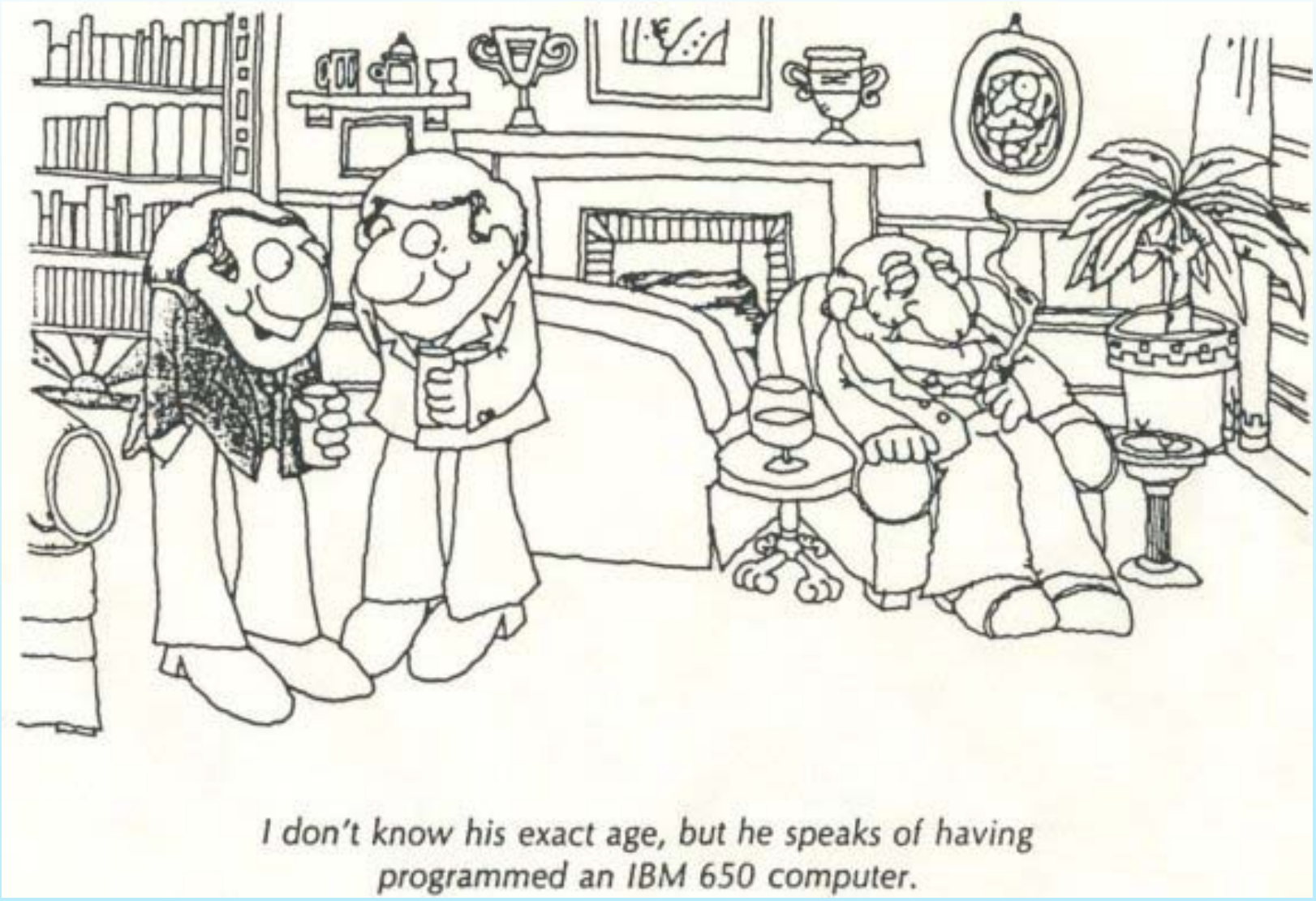
Agenda for Talk

- Short Tour of Computer Architecture History
- Current State of Computer Architecture and Its Effect on Code Performance
- Projections About the Future

In the Beginning

- **Strict Serial von Neumann architecture with one processor, one path to memory, no hardware threads.**
- **1954: IBM 650**
  - Memory: 2000 words (10-bit) on rotating drum
  - Processor: add, subtract, multiply, divide, branch, table look-up



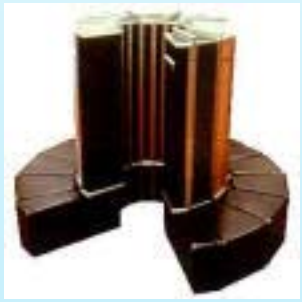


I don't know his exact age, but he speaks of having programmed an IBM 650 computer.

Then Came Seymour Cray

- **1964: CDC 6600. Functional parallelism with 10 functional units (add, multiply, divide, etc.). Multi-threading with 10 peripheral processing units.**

- **1974: Cray 1. Introduced vectors with 8 vector registers, each holding 64 words. Scatter-gather instructions for non-continuous stride vectors. Continued functional parallelism.**



Then Came Seymour Cray (cont.)

- 1985: Cray XMP. Four Cray 1 processors in parallel, shared memory. (Major contributions from Steve Chen)

- 1985: Cray 2. Eight processors. 2 GB shared CMOS memory. 64 or 128-way memory interleave. 128 KB cache. Eight 16-word prefetch buffers.

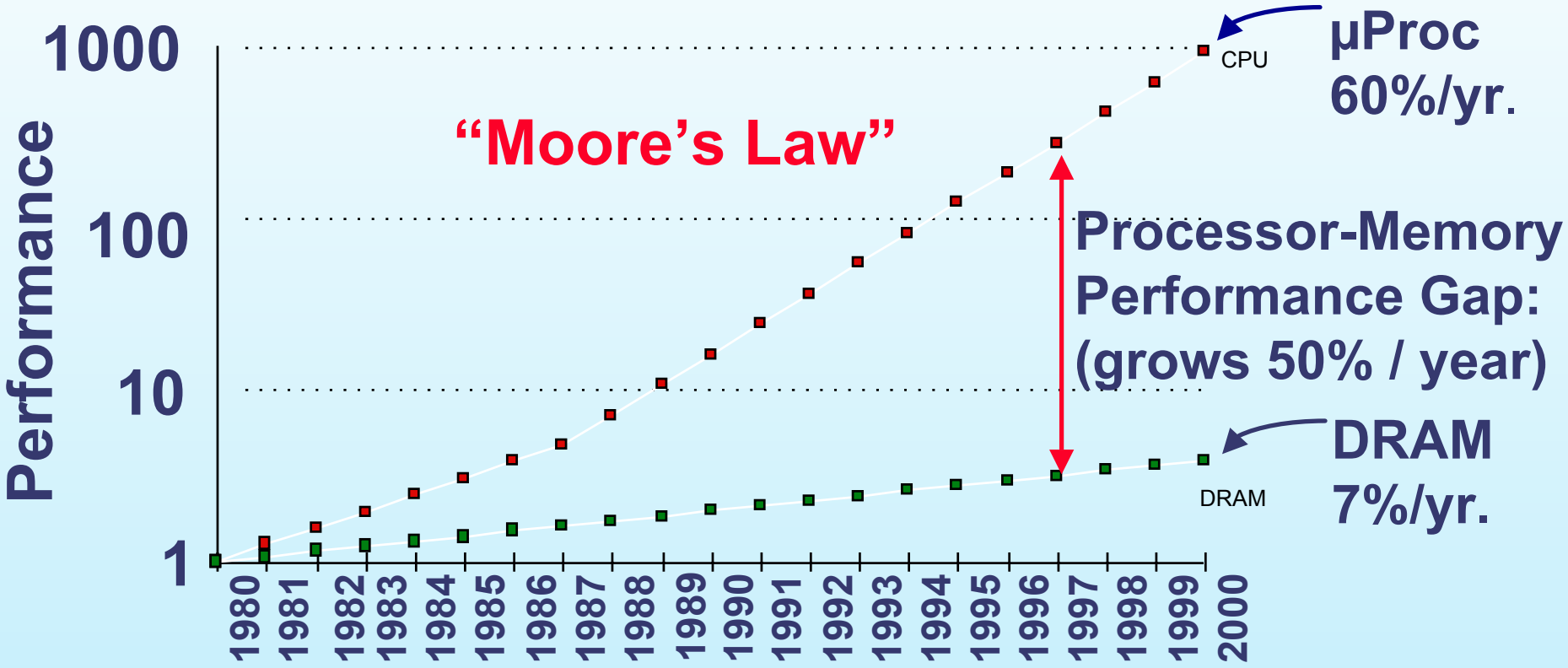
Then Came the Flood (Late 1980s and Early 1990s)

- **Many versions of parallelism, e.g.**
  - **Massively parallel MIMD**: nCube, Intel Paragon, Thinking Machines CM-5, Cray T3D
  - **SIMD**: MasPar, Thinking Machines CM-1
  - **NUMA**: Kendall Square KSR-1
  - **Multi-threading**: Denelcor HEP
  - **Beowulf** commodity clusters
- **Spurred by Moore's law, killer micros, limit of speed of light**
- **However**
  - Processors got faster, memory didn't
  - Interconnects added latency and limited bandwidth
  - Code for parallel machines hard to program, debug, and optimize

Major Problem: Poor Links Between Workload and Architecture Design

- **Build It and They Will Come**
- **Weakness of Government High Performance Computing and Communication Program in 1990s**
  - No link between grants for computer architectures and grants for computer acquisition
  - Poor feedback from users to developers
  - Poor connections between computational scientists and computer scientists (one workshop in Pittsburgh in 1993)
- **Result: Selection of computer architecture is more religion than science**

Processor-Memory Performance Gap



- Alpha 21264 full cache miss / instructions executed:
  180 ns/1.7 ns =108 clks x 4 or 432 instructions
- Caches in Pentium Pro: 64% area, 88% transistors

*Taken from Patterson-Keeton Talk to SigMod

Processing vs. Memory Access

- **Doesn't cache solve this problem?**
  - It depends. With small amounts of contiguous data, usually. With large amounts of non-contiguous data, usually not.
  - In most computers the programmer has no control over cache.
  - Often "a few" Bytes/FLOP is considered OK.

- **However, consider operations on the transpose of a matrix (e.g., for adjunct problems)**

  $Xa = b$ $\qquad\qquad$ $X^T a = b$

  - If X is big enough, 100% cache misses are guaranteed, and we need at least 8 Bytes/FLOP (assuming a and b can be held in cache).

- **Latency and limited bandwidth of processor-memory and node-node communications are major limiters of performance for scientific computation**

Testing Processing vs. Memory Access with Benchmarks

- **Simple benchmark: Stream Triad**

$$a_i + s \times b_i = c_i$$

a_i, b_i, and c_i are vectors; s is a scalar. Vector length is chosen to be much longer than cache size.

Each execution includes 2 memory loads + 1 memory store and 2 FLOPs, or 12 Bytes/FLOP (assuming 8 Byte precision)

No computer has enough memory bandwidth to reference 12 Bytes for each FLOP!

Testing Processing vs. Memory Access with Benchmarks

- **Another Benchmark: Linpack**

$$A_{ij}\, x_j = b_i$$

Solve this linear equation for the vector x, where A is a known matrix, and b is a known vector. Linpack uses the BLAS routines, which divide A into blocks.

On the average Linpack requires 1 memory reference for every 2 FLOPs, or 4Bytes/Flop.

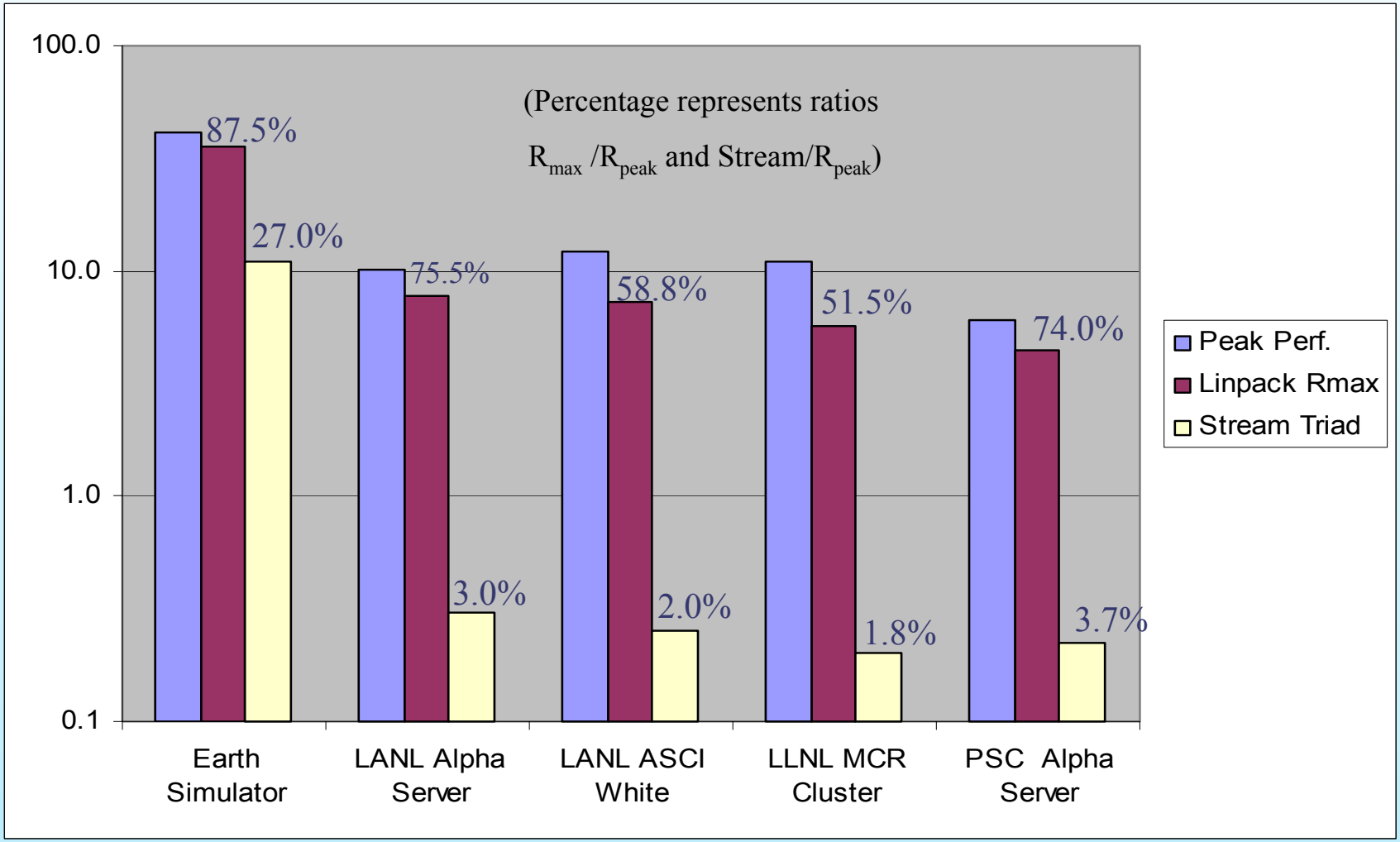
Many of these can be cache references.

Selected System Characteristics

		Earth Simulator (NEC)	ASCI Q (HP ES45)	ASCI White (IBM SP3)	MCR (Dual Xeon)	32 Node Cray X1 (Cray)
Year of Introduction		2002	2003	2000	2002	2003
Node Architecture		Vector	Alpha micro	Power 3 micro	Xeon micro	Vector
		SMP	SMP	SMP	SMP	SMP
System Topology		NEC single-stage	Quadrics QsNet	IBM	Quadrics QsNet	2D Torus
		Crossbar	Fat-tree	Omega network	Fat-tree	Interconnect
Number of Nodes		640	2048	512	1152	32
Processors	- per node	8	4	16	2	4
	- system total	5120	8192	8192	2304	128
Processor Speed		500 MHz	1.25 GHz	375 MHz	2.4 GHz	800 MHz
Peak Speed	- per processor	8 Gflops	2.5 Gflops	1.5 Gflops	4.8 Gflops	12.8 Gflops
	- per node	64 Gflops	10 Gflops	24 Gflops	9.6 Gflops	51.2 Gflops
	- system total	40 Tflops	30 Tflops	12 Tflops	10.8 Tflops	1.6 Tflops
Memory	- per node	16 GB	16 GB	16 GB	16 GB	8-64 GB
	- per processor	2 GB	4 GB	1 GB	2 GB	2-16 GB
	- system total	10.24 TB	48 TB	8 TB	4.6 TB	
Memory Bandwidth (peak)						
	- L1 Cache	N/A	20 GB/s	5 GB/s	20 GB/s	76.8 GB/s
	- L2 Cache	N/A	13 GB/s	2 GB/s	1.5 GB/s	
	Main (per processor	32 GB/s	2 GB/s	1 GB/s	2 GB/s	34.1 GB/s
Inter-node MPI						
	- Latency	8.6 µsec	5 µsec	18 µsec	4.75 µsec	8.6 µsec
	- Bandwidth	11.8 GB/s	300 MB/s	500 MB/s	315 MB/s	11.9 GB/s
Bytes/flop to main memory		4	0.8	0.67	0.4	3
Bytes/flop interconnect		1.5	0.12	0.33	0.07	1

Most of this data is from Kerbyson, Hoisie, Wasserman; LANL; unpublished. Additional data from Jack Dongarra.

Performance Measures of Selected Top Computers



Note Logarithmic Y axis

What About Synthetic Benchmarks?

- Peak performance – nuf said
- Linpack –only measures performance of cache-friendly code
- Stream – only measures contiguous communications with memory, but good measure of bandwidth
- GUPS – really tough benchmark because it makes random memory access; may exceed requirements of most codes
- IDC balanced benchmarks – good compilation, but somewhat artificially combined
- Effective System Performance Benchmark – promising, but not widely used
- NAS Parallel Benchmarks – disused, but may be coming back
- Livermore Loops – obsolete
- Your own workload - ??

Resurgence of Performance Analysis Is Promising

- LANL Performance and Architecture Lab: http://www.c3.lanl.gov/par_arch/
- Performance Evaluation Research Center: http://perc.nersc.gov/
- IDC User Forum: http://64.122.81.35/benchmark/
- Performance Modeling and Characterization: http://www.sdsc.edu/PMaC/Benchmark/
- NAS Parallel Benchmarks: http://www.nas.nasa.gov/Software/NPB/
- Recent High End Computing Workshop offered recommendations for performance evaluation: http://www.cra.orgActivities/workshops/nitrd/
- Great opportunity for agencies to cooperate on performance evaluation.

Trade-Offs Between Commodity Clusters and Custom Supercomputers

Clusters

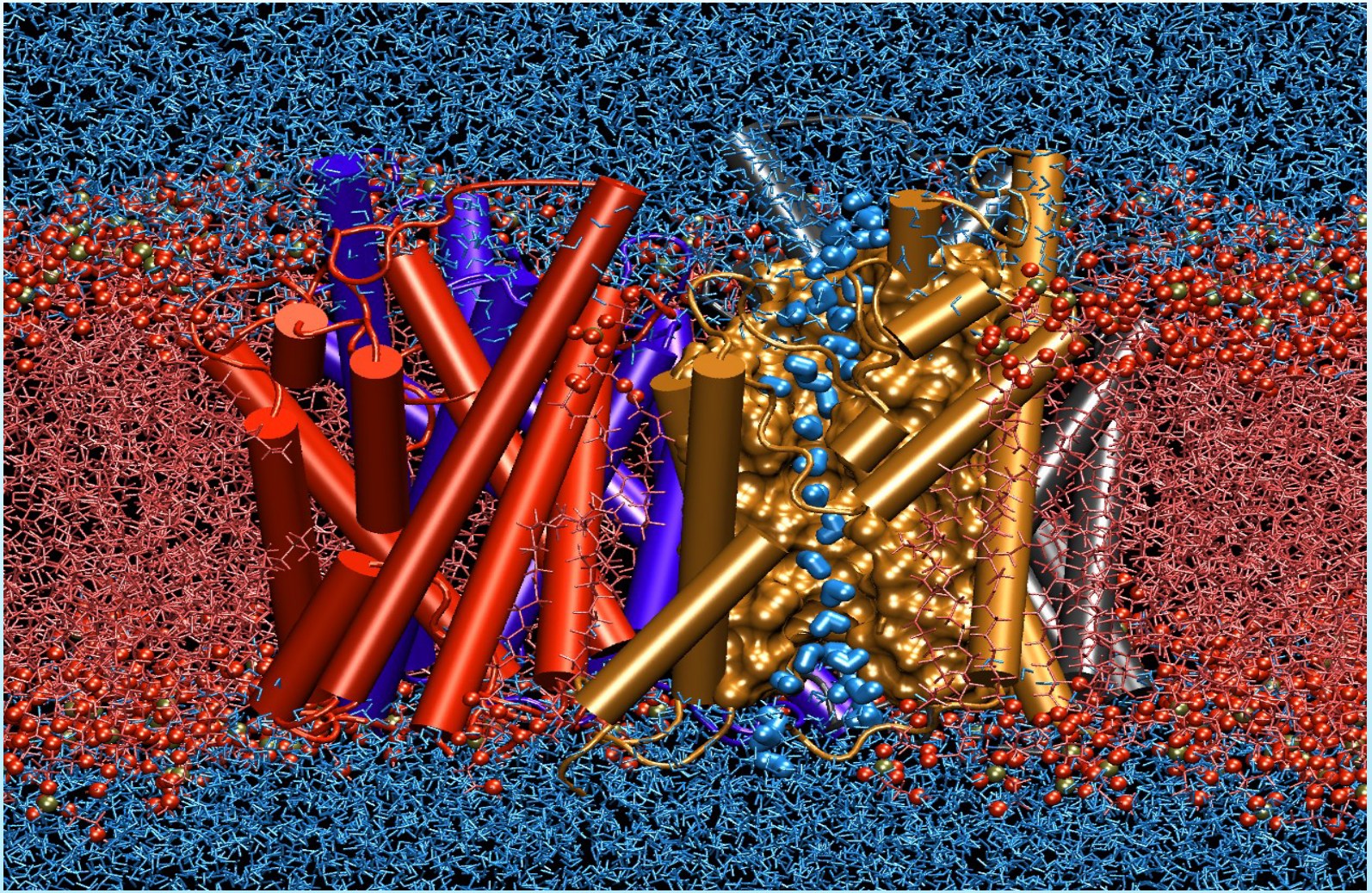
- Absolutely Cheap
- Cheap for peak FLOPS/$
- Low direct maintenance $
- Large volume per node
- High power requirements per node
- Easy to develop code on workstations
- Efficient for code with limited communication that fits in cache
- Benchmark to be sure!

Supercomputers

- Absolutely Expensive
- May be cheap for sustained FLOPS/$
- Smaller volume per node
- Lower power requirements per node
- Harder to develop code on workstations
- Efficient for large codes with high communications requirements
- Benchmark to be sure!

Simulation of Aquaporin Protein Inside a Cell (PSC Alpha Cluster)

Visualization shows transport of water molecules into cell.



For Further Information

Please contact us at:

nco@itrd.gov

Or visit us on the Web:

www.itrd.gov